1   21.   (new)   A method of mapping a plurality of virtual

2   registers to a plurality of physical registers, the method

3   comprising:

4   providing a plurality of virtual registers, wherein each of

5   the virtual registers comprises physical register

6   address bits; and providing a status indicator for

7   indicating a status of each of the virtual registers;

8   designating a subset of the virtual registers as virtual

9   local registers;

10   executing a save command, the executing of the save command

11   comprising saving a mapping of all of the virtual

12   local registers onto a stack; saving a status as

13   indicated by the status indicator for each of the

14   virtual local registers onto the stack, and setting

15   the status of all of the virtual local registers to

16   "clean";

17   executing a restore command, the executing the restore

18   command comprising popping the mapping of all of the

19   virtual local registers from the stack to the virtual

20   local registers; and popping the status of all of the

21   virtual local registers from the stack;

22   binding an argument, the argument binding comprising binding

23   a first virtual register of the virtual registers to a

24   second virtual register of the virtual registers; and

25   binding the status of the first virtual register to

26   the second virtual register;

27   wherein the argument binding further comprises saving a

28   mapping of the second virtual register onto the stack,

29   saving the status of the second virtual register onto

30   the stack, placing a physical address stored in the

31   first virtual register in the second virtual register,

32   and setting the status of the second virtual register

33   to the status of the first virtual register; and

34   wherein the argument binding occurs during a call

35   instruction, wherein the call instruction has at least

36   one argument, wherein the first virtual register is

37   used for the at least one argument.

1 22.   (new) A method comprising:

2   decoding an instruction;

3   if the instruction is a call instruction, then binding an

4   argument of the call instruction;

5   wherein the argument binding comprises

6   copying a first virtual register of a plurality of

7   virtual registers to a second virtual register of

8   the virtual registers, each of the virtual

9   registers comprising physical register address

10   bits, and

11   copying a first status indicator of a plurality of

12   status indicators to a second status indicator of

13   the status indicators, wherein each of the status

14   indicators corresponds to a respective one of the

15   virtual registers; and

16   wherein the first virtual register is used for the argument

17   and the second virtual register is used as a formal

18   parameter.

1 23.   (new)   The method, as recited in claim 22, further

2 comprising:

3   mapping a virtual register of the plurality of virtual

4   registers from an old physical register to a new

5   physical register, when the virtual register is a

6   destination virtual register of an instruction being

7   decoded; and

8   placing an address of the old physical register in an

9   instruction retirement list related to the instruction

10   being decoded if and only if the status indicator

11   corresponding to the virtual register is not clean.

1 24.   (new)   The method, as recited in claim 23, further

2 comprising:

3   saving the physical register address bits held in the second

4   virtual register and the second status indicator to a

5   stack and then setting to clean the second status

6   indicator.

1 25.   (new)   The method, as recited in claim 23, further

2 comprising:

3   setting the second status indicator to not clean when the

4   second virtual register is mapped to a new physical

5   register.

1 26.   (new)   The method, as recited in claim 22, wherein:

2   a subset of the plurality of virtual registers are virtual

3   local registers.

1 27.   (new)   The method, as recited in claim 26, further   ,

2 comprising:

3   executing a save command, the executing of the save command

4   comprising saving the mapping of all of the virtual

5   local registers onto a stack; and saving the status

6   indicators corresponding to all of the virtual local

7   registers onto the stack.

1 28.   (new)   The method, as recited in claim 27, wherein:

2   the executing of the save command further comprises setting

3   the status indicators corresponding to all of the

4   virtual local registers to clean after the saving of

5   the status indicators onto the stack.

Docket#: WDGNP001
Serial No.: 09/825,753

1 29.   (new) The method, as recited in claim 28, further

2 comprising:

3   executing a restore command, the executing the restore

4   command comprising popping the mapping of all of the

5   virtual local registers from the stack to the virtual

6   local registers; and popping the status indicators

7   corresponding to all of the virtual local registers

8   from the stack.

1 30.   (new) The method, as recited in claim 29, further

2 comprising:

3   selectively executing the restore command if the instruction

4   is a return instruction.

1 31.   (new) The method, as recited in claim 22, wherein:

2   following the argument binding, if the first virtual

3   register is a destination register, the first virtual

4   register is assigned a first physical register address

5   which is different than a second physical register

6   address stored in the second virtual register.

Docket#: WDGNP001
Serial No.: 09/825,753

1 | 32.   (new)   The method, as recited in claim 31, wherein:

2 | before the assignment of the first physical register address

3 | to the first virtual register, a corresponding first

4 | physical register status is "free".

1 | 33.   (new)   The method, as recited in claim 32, wherein:

2 | after the assignment of the first physical register address

3 | to the first virtual register, the corresponding first

4 | physical register status is "waiting".

1   34.   (new)   A processing device including:

2   an instruction decoder adapted to decode an instruction;

3   a plurality of physical registers;

4   a plurality of virtual registers, each of the virtual

5   registers comprising physical register address bits;

6   a plurality of status indicators, each of the status

7   indicators corresponding to a respective one of the

8   virtual registers;

9   wherein if the instruction decoder decodes a call

10   instruction, then binding an argument of the call

11   instruction, the binding comprising

12   copying a first one of the virtual registers into a

13   second one of the virtual registers, and

14   copying a first one of the status indicators into a

15   second one of the status indicators, the first

16   status indicator corresponding to the first

17   virtual register and the second status indicator

18   corresponding to the second virtual register; and

19   wherein the first virtual register is used for the argument

20   and the second virtual register is used for a formal

21   parameter.

1   35.   (new)   The processing device, as recited in claim 34,

2   wherein:

3   if the instruction decoder decodes an instruction having a

4   destination virtual register selected from the virtual

5   registers, then

6   mapping the destination virtual register from an old ·

7   physical register of the physical registers to a

8   new physical register of the physical registers,

9   and

10   placing an address of the old physical register in an

11   instruction retirement list related to the

12   instruction if and only if the status indicator

13   corresponding to the destination virtual register

14   is not clean.

1   36.   (new)   The processing device, as recited in claim 35,

2   further including:

3   a stack; and

4   wherein if the instruction decoder decodes a call

5   instruction, then saving the physical register address

6   bits held in the second virtual register and the

7   second status indicator to the stack and then setting

8   to clean the second status indicator.

14 of 30

1   37.   (new)   The processing device, as recited in claim 35,

2   wherein:

3   when the second virtual register is mapped to a new physical

4   register, setting the second status indicator to not

5   clean.

1   38.   (new)   The processing device, as recited in claim 34,

2   wherein:

3   a subset of the plurality of virtual registers are virtual

4   local registers.

1   39.   (new)   The processing device, as recited in claim 38,

2   further including:

3   a stack; and

4   wherein execution of a save command comprises saving the

5   mapping of all of the virtual local registers onto the

6   stack; and saving the status indicators corresponding

7   to all of the virtual local registers onto the stack.

1   40.   (new)   The processing device, as recited in claim 39,

2   wherein:

3   the execution of the save command further comprises setting

4   the status indicators corresponding to all of the

5   virtual local registers to clean after the saving of

6   the status indicators onto the stack.

1   41.   (new)   The processing device, as recited in claim 40,

2   wherein:

3   execution of a restore command comprises popping the mapping

4   of all of the virtual local registers from the stack

5   to the virtual local registers; and popping the status

6   indicators corresponding to all of the virtual local

7   registers from the stack.

1   42.   (new)   The processing device, as recited in claim 41,

2   wherein:

3   if the instruction decoder decodes a return instruction,

4   then executing the restore command.

1   43.   (new)   The processing device, as recited in claim 34,

2   wherein:

3   following the argument binding, if the first virtual

4   register is a destination register, the first virtual

5   register is assigned a first physical register address

6   which is different than a second physical register

7   address stored in the second virtual register.

1   44.   (new)   A method comprising:

2   decoding an instruction;

3   maintaining a mapping of virtual registers to physical

4   registers, a subset of the virtual registers being

5   virtual local registers;

6   if the instruction is a save instruction, then executing a

7   save command;

8   if the instruction is a restore instruction, then executing

9   a restore command; and

10   wherein

11   the executing of the save command comprises saving the

12   mapping of all of the virtual local registers

13   onto a stack, and

14   the executing of the restore command comprises popping

15   the mapping of all of the virtual local registers

16   from the stack to the virtual local registers.

1   45.   (new)   The method, as recited in claim 44, wherein:

2   the executing of the save command further comprises saving

3   status indicators corresponding to all of the virtual

4   local registers onto the stack, and

5   the executing of the restore command further comprises

6   popping the status indicators corresponding to all of

7   the virtual local registers from the stack.

1   46.   (new)   The method, as recited in claim 45, wherein:

2   the executing of the save command further comprises setting

3   the status indicators corresponding to all of the

4   virtual local registers to clean after the saving of

5   the status indicators corresponding to all of the

6   virtual local registers onto the stack.

1   47.   (new)   A processing device including:

2   an instruction decoder adapted to decode an instruction;

3   a plurality of physical registers;

4   a plurality of virtual registers, each of the virtual

5   registers comprising physical register address bits,

6   and a subset of the virtual registers being virtual

7   local registers; and

8   wherein

9   if the instruction decoder decodes a save instruction,

10   then executing a save command, the executing of

11   the save command comprising saving a mapping of

12   all of the virtual local registers onto a stack,

13   and

14   if the instruction decoder decodes a restore

15   instruction, then executing a restore command,

16   the executing of the restore command comprising

17   popping the mapping of all of the virtual local

18   registers from the stack to the virtual local

19   registers.

1 48.   (new)   The processing device, as recited in claim 47,

2 further including:

3   a plurality of status indicators, each of the status

4   indicators corresponding to a respective one of the

5   virtual registers; and

6   wherein

7   the executing of the save command further comprises

8   saving the status indicators corresponding to all

9   of the virtual local registers onto the stack,

10   and

11   the executing of the restore command further comprises

12   popping the status indicators corresponding to

13   all of the virtual local registers from the

14   stack.

1 49.   (new)   The processing device, as recited in claim 48,

2 wherein:

3   the executing of the save command further comprises setting

4   the status indicators corresponding to all of the

5   virtual local registers to clean after the saving of

6   the status indicators corresponding to all of the

7   virtual local registers onto the stack.

20 of 30

Docket#: WDGNP001
Serial No.: 09/825,753

1 50.   (new)   The processing device, as recited in claim

2   49, further including:

3   ·   ·a plurality of physical register status indicators, each of

4   the physical register status indicators corresponding

5   to a respective one of the physical registers; and

6   wherein each of the physical register status indicators

7   represents a selected one of a plurality of physical

8   'register states, and the physical register states

9   include "free", "waiting", and "valid".

1 51.   (new)   The processing device, as recited in claim

2   50, wherein:

3   physical registers available for mapping to virtual

4   registers are represented as "free" in the

5   corresponding physical register status indicators.

1 52.   (new)   The processing device, as recited in claim

2   51, wherein:

3   physical register status indicators transition to

4   representing "waiting" when the corresponding physical

5   registers are mapped to virtual registers.

21 of 30

1 53.   (new)   The processing device, as recited in claim

2 52,wherein:

3   physical register status indicators transition to

4   representing "valid" when the corresponding physical

5   registers are written.

22 of 30